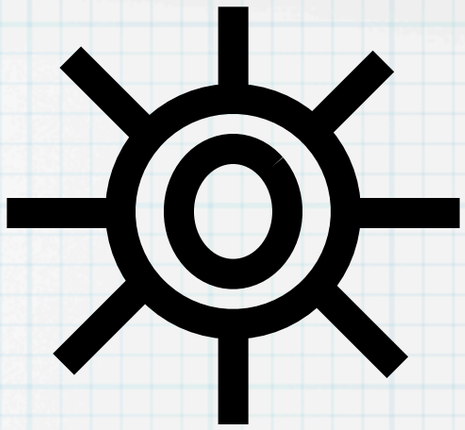
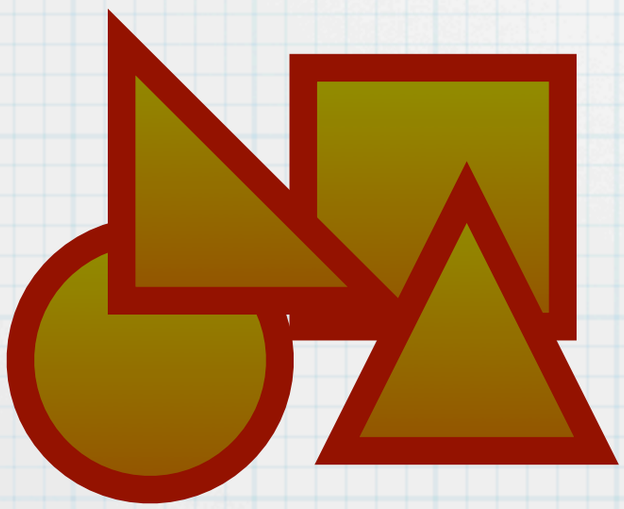


"The World of Relations"

an ontology of the relational paradigm

Les Waguespack, Ph.D.



Copyright and References

* Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377-387

* Codd, E.F. (1990). The Relational Model for Database Management, Version 2, Addison Wesley Publishing Company. ISBN 0-201-14192-2.

* Database Systems, Connolly & Begg, Addison-Wesley, ISBN: 0-321-21025-5.

What is an ontology?

* ontology: the branch of metaphysics dealing with the nature of being new oxford dictionary

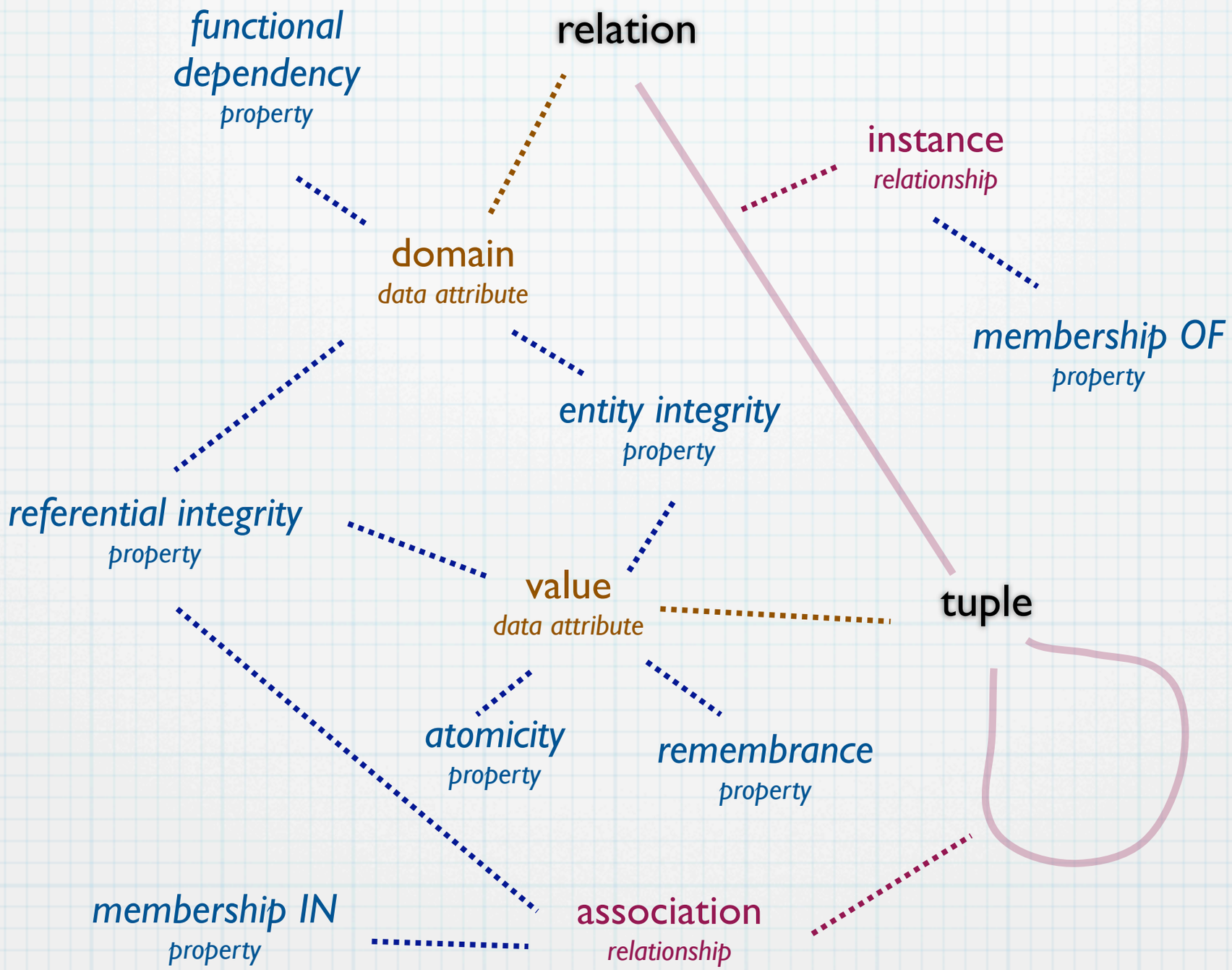
* metaphysics: the branch of philosophy that deals with the first principles of things, including abstract concepts such as being, knowing, substance, cause, identity, time, and space.

* "what exists, how do we understand it, what explains it, what does it explain?"

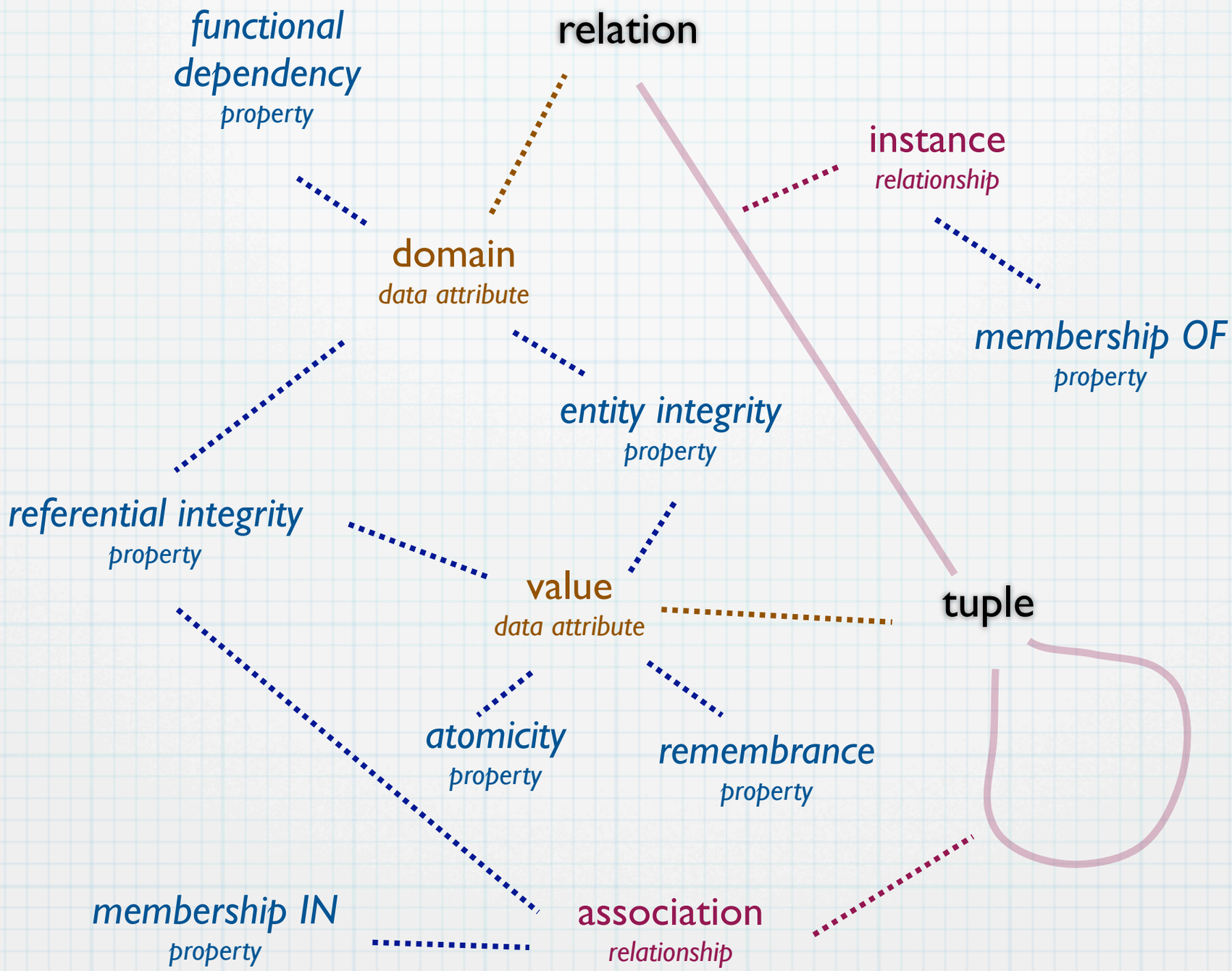
What does an ontology do for us?

* It helps us describe the "world!"
  * a common terminology shared by the community
  * shared rationale explaining properties
* What questions does an ontology answer?
  * what are the things? - "individuals"
  * how are they described? - "attributes"
  * what things go together? - "classes"
  * how do things relate to one another? - "relationships"

Relational Ontology - Graphically



Once over quickly!



What things are in OO?

* Relational Paradigm: Individual - **tuple**
  * derived from the living physical experience of humans seeing and touching things - projected onto non-concrete abstractions as well
* tuples are distinct
  * they are separable by nature of how they are described,
  * they are distinguishable because of their attribute values,
  * tuples are sometimes called **rows**
  * the set of all data attribute values in a single attribute for all tuples in a relation is sometimes called a **column**

What describes relational things?

* Relational Paradigm: Attributes - Attributes
  * Relational "things" are described by their **attributes**
  * Data attribute values are **atomic** - indivisible and whole

attributes	data	behavioral
static	**data attributes domains** define what "can" be stored and recalled: the property of **remembrance**	the **functional dependency** of one attribute value on another expresses **intension**
dynamic	**data attributes values** possess what "is" stored and recalled: the property of **remembrance**	adherence to **intension** in the actual values in tuples (**extension**) realizes **integrity**

What things go together?

* Relational Paradigm: Classification - **relation**
* The relationship **instance of** defines a collection of tuples individually expressing specialized instances of the same concept (A relations is sometimes called a **table**)
  * Every tuple is an **instance of** its relation and shares the same static structure defined by that relation with every other tuple of that relation
  * Tuples are said to be "**members of** their relation."
* Relation structure - data
  * Static data attributes are defined in the relation's attribute **domains**

How do things relate?

* Relational Paradigm: Relationships
  - (structural and behavioral)
    * every tuple in a relation shares the identical structure of attributes and domains
    * although the ordering of attributes is immaterial in a relation their ordering among all tuples is consistent
    * integrity properties defined in the relation apply to all tuples of the relation

How do things relate? (cont.)

* Relational Paradigm: Relationships (continued)
  * behavioral- (functional dependency, entity integrity, association, referential integrity)
    * **functional dependency** relates attributes - prescribes that the value of one attribute(s) will **determine** the value of another(s) in a tuple for all tuples in a relations
    * **entity integrity** stipulates that exists a subset of the attribute(s) in a relation whose value(s) uniquely discriminates each tuple within a relation and that no attribute(s) in this subset may be **null** (of indeterminate value) A subset of attributes of this type is called a **candidate key** one of which is designated as the **primary key**!)

How do things relate? (cont..)

* Relational Paradigm: Relationships (continued)
  * behavioral- (functional dependency, entity integrity, association, referential integrity)
    * **association** relates tuples across two relations (or a relation with itself) - data attribute values in attributes of the two relations from the same domain (**join compatibility**) define a set of tuples in one relation as related to one tuple in the other - the property of **membership IN** the association (membership **IN** a group is distinct from member **of** a relation)

How do things relate? (cont...)

- **association** (continued)
  - **Relational Operations** - Membership IN is realized through relational operations keying on relation structure and values. Each relational operation produces a real or virtual relation as its result. The **selection** operation retrieves tuple(s) (also called **row(s)**) based upon a **selection predicate** testing data attribute value(s) to determine whether each tuple pair is or is not in the set. Selection predicates are based on any boolean comparison including constant values or values referenced in data attribute value(s). The **projection** operation copies all the data attribute value(s) for a particular attribute (also called a **column**).

How do things relate? (cont...)

* **association** (continued)
  * Association between relations (or a relation and itself) is based upon relating (matching) data attribute values in tuples of one relation with those of another. The **join** operation pairs every combination of tuples from one relation with those of another relation and copies the data attribute values from the pairs into the result where the pairing satisfies a **selection predicate**. This relational operation is called join because facts from two sources are joined in the result.

How do things relate? (cont...)

* **association** (continued)
  * **Join compatibility** requires that the values involved in comparisons (i.e. selection predicates) whether constants or data attribute values derive from the same data **attribute domain**.
  * **Referential Integrity:** when relations are devised such that a tuple in one relation predisposes the existence of (**owns**) tuple(s) in another, the data attribute(s) of the second required to join the relations is called a **foreign key**. Referential integrity asserts that any value found in the data value attribute(s) of a foreign key must appear in a tuple of the first relation as the value of a candidate key or itself be **null** (of indeterminate value).

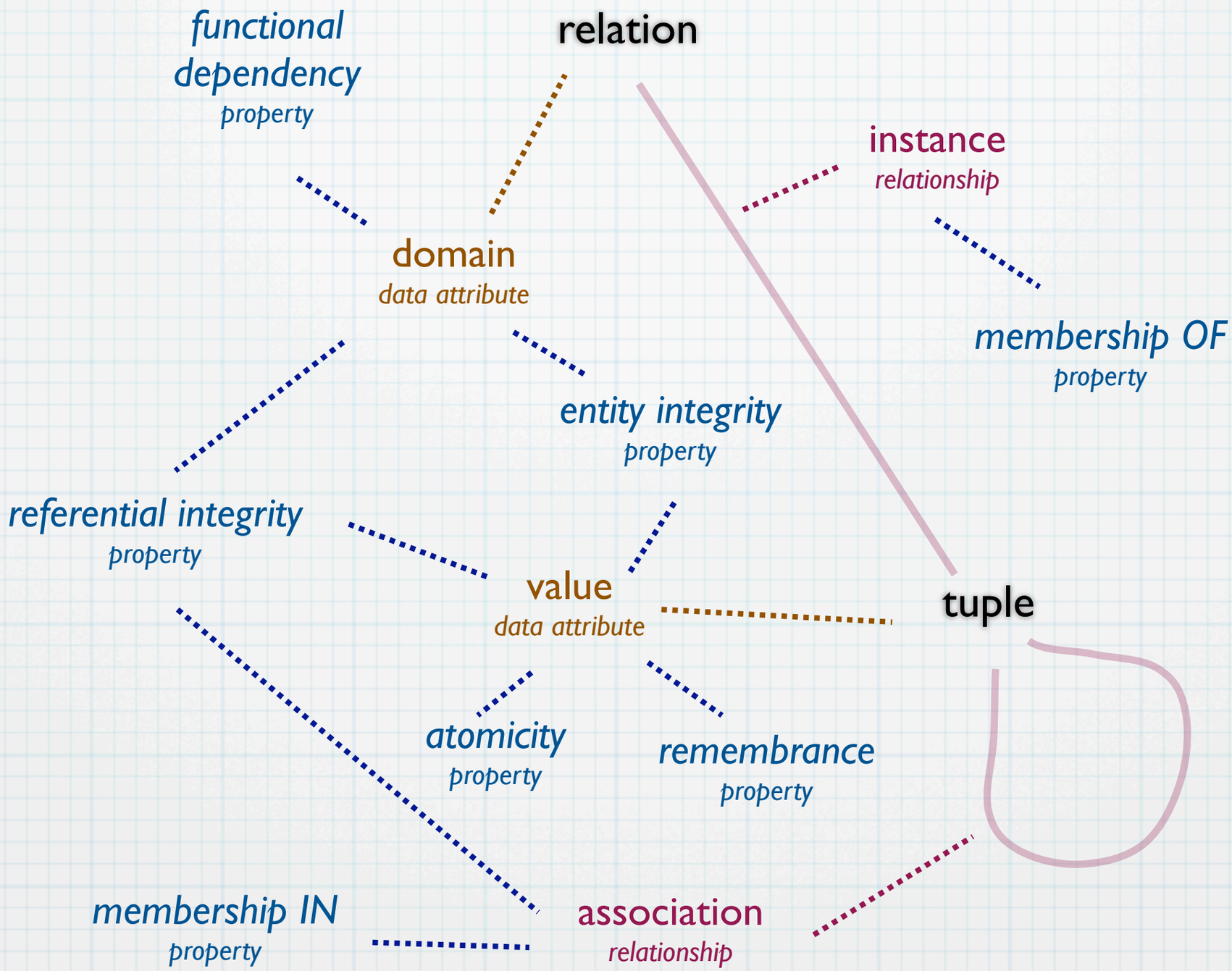
How do things relate? (cont...)

* **Normalization**: relational model consistency depends on the semantic concurrence of the behavioral relationships and the objectives of the database modeler, the **intension**, (rather than the accident of a relation's contents at any particular instant, its **extension**). The integrity properties defined above enable the database modeler to devise a structure and behavior of relations that avoid semantic discord called **anomalies**, the unintended loss or modification of information by relational operations. Relations designed to avoid certain kinds of anomalies are said to be **normalized** or in **normal form**. Normalization is the arrangement of data attributes and their relationships among relation structures to prevent particular anomalies.

How do things relate? (cont...)

- **Normalization** (continued)
  - **First Normal Form** asserts that every data attribute value is **atomic**, indivisible in value or form and may not be operated upon except as a whole and single value.
  - **Second Normal Form** is first normal form and asserts that every data attribute value not in the primary key is **fully functionally dependent** upon the primary key. ("Fully" means applying to every data attribute of the primary key.)
  - **Third Normal Form** presupposes first and second normal forms and asserts that **no** data attribute outside the primary key is **transitively** dependent upon the primary key. ("Transitively" means an attribute(s) functionally dependent upon an attribute functionally dependent upon an attribute (. . .) functionally dependent on the primary key.)

One More Time!



You Need to be able to Explain:

- The Relational Ontology
  - Individuals
    - Tuple
  - Attributes
    - Data Attribute
  - Classes
    - Relation
  - Relationships
    - Behavioral Relationships
    - Functional Dependency
    - Entity Integrity
    - Association
    - Relational Operations
      - Selection
      - Projection
      - Join
    - Join Compatibility
    - Referential Integrity
- Normalization
  - First Normal Form
  - Second Normal Form
  - Third Normal Form

Relational Ontology - Graphically